Laporan Hasil Praktikum

Algoritma Dan Struktur Data

Jobsheet 14



Angel Chelssa Leoniy Eka Permatasari

244107020202

1E

Program Studi Teknologi Informasi

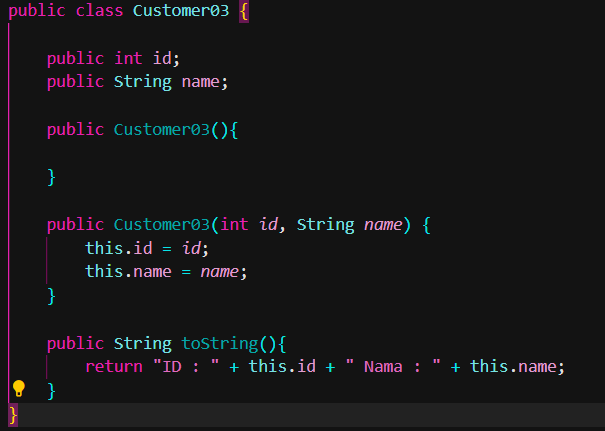
Jurusan Teknik Informatika

POLINEMA

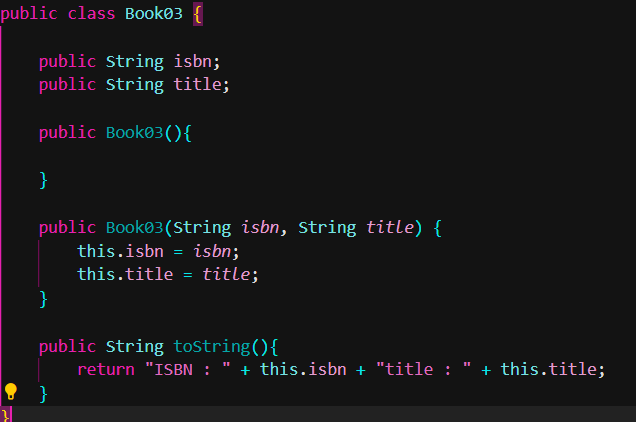
2025

**PERSIAPAN**

1. Buat folder Praktikum14
2. Buat class Customer dalam Customer.java

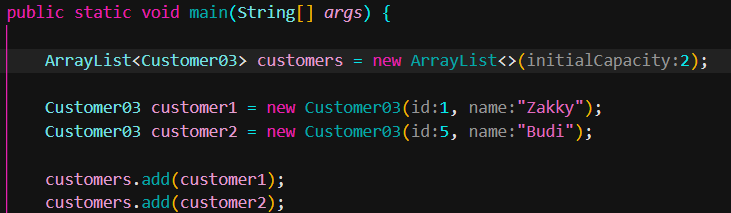


1. Buat class Book dalam Book.java

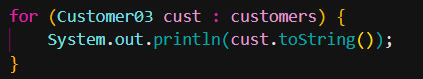


**PERCOBAAN 1**

1. Buat file DemoArrayList.java. Lakukan import java.util.ArrayList;
2. Pada fungsi main(), instansiasi collection baru dengan nama customers bertipe ArrayList of Customer dengan size 2. Selanjutnya, buat object customer1 dan customer2 kemudian tambahkan ke dalam ArrayList customers dengan method add.



1. Gunakan looping dengan foreach untuk mencetak data customers



1. Cobalah tambahkan object customer baru ke dalam customers. Apakah object dapat ditambahkan meskipun melebihi kapasitas?



1. Compile dan run kode program, di mana object yang baru ditambahkan? Di awal, di tengah, atau di akhir collection?

**Jawab:**

Object "Cica" ditambahkan di akhir karena method .add() tanpa indeks.

1. Untuk menambahkan object baru pada index tertentu, lakukan sebagai berikut



1. Compile dan run kode program. Index pada ArrayList dimulai dari 0 atau 1?

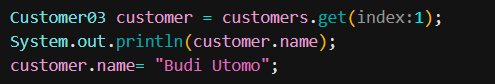
**Jawab:**

Index pada ArrayList dimulai dari 0

1. Untuk mengetahui posisi dari suatu objek, gunakan method indexOf()



1. Untuk mengembalikan object pada index tertentu, gunakan method get()

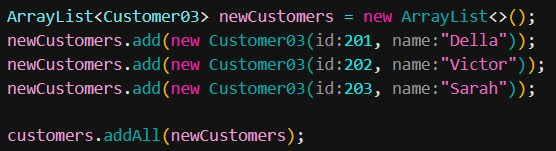


1. Cobalah hapus angka 2 saat instansiasi object customers. Apakah ArrayList dapat diinstansiasi tanpa harus menentukan size di awal?

**Jawab:**

Ya, ArrayList dapat diinstansiasi tanpa harus menentukan size di awal

1. Anda juga dapat menambahkan sekumpulan customer baru ke dalam ArrayList secara sekaligus. Misalnya terdapat ArrayList newCustomers. Tambahkan seluruh object customer sekaligus ke dalam customers.



1. Karena sudah menyediakan method toString(), pengecekan data customers untuk proses debugging juga dapat dilakukan lebih sederhana dengan cara berikut.



**KODE PROGRAM**

**KODE PROGRAM CLASS Customer03.java**

|  |
| --- |
| public class Customer03 {      public int id;      public String name;      public Customer03(){      }      public Customer03(int *id*, String *name*) {          this.id = *id*;          this.name = *name*;      }      public String toString(){          return "ID : " + this.id + " Nama : " + this.name;      }  } |

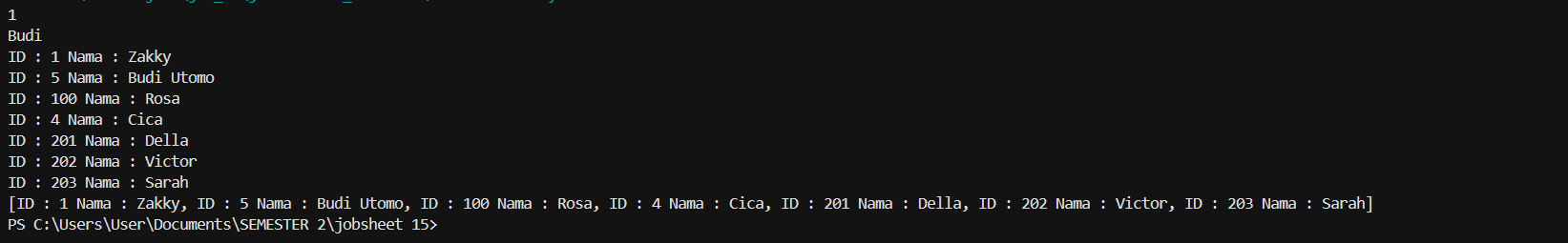
**KODE PROGRAM CLASS Book03.java**

|  |
| --- |
| public class Book03 {      public String isbn;      public String title;      public Book03(){      }      public Book03(String *isbn*, String *title*) {          this.isbn = *isbn*;          this.title = *title*;      }      public String toString(){          return "ISBN : " + this.isbn + " title : " + this.title;      }  } |

**KODE PROGRAM CLASS DemoArrayList03.java**

|  |
| --- |
| import java.util.ArrayList;  public class DemoArrayList03 {      public static void main(String[] *args*) {            ArrayList<Customer03> customers = new ArrayList<>();          Customer03 customer1 = new Customer03(1, "Zakky");          Customer03 customer2 = new Customer03(5, "Budi");          customers.add(customer1);          customers.add(customer2);          customers.add(new Customer03(4, "Cica"));          customers.add(2, new Customer03(100, "Rosa"));          System.out.println(customers.indexOf(customer2));          Customer03 customer = customers.get(1);          System.out.println(customer.name);          customer.name= "Budi Utomo";          ArrayList<Customer03> newCustomers = new ArrayList<>();          newCustomers.add(new Customer03(201, "Della"));          newCustomers.add(new Customer03(202, "Victor"));          newCustomers.add(new Customer03(203, "Sarah"));          customers.addAll(newCustomers);          for (Customer03 cust : customers) {              System.out.println(cust.toString());          }            System.out.println(customers);      }  } |

**HASIL KODE**

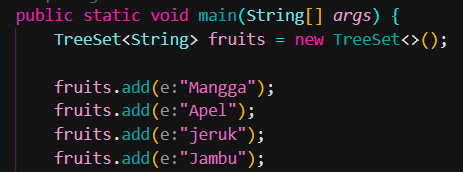
****

**Praktikum - Implementasi TreeSet**

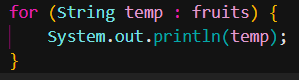
1. Buat file TreeSetDemo.java kemudian import java.util.TreeSet;



1. Tambahkan fungsi main() kemudian instansiasi object TreeSet of String. Tambahkan beberapa nilai bertipe String ke dalam TreeSet



1. Cetak data pada ts dengan looping

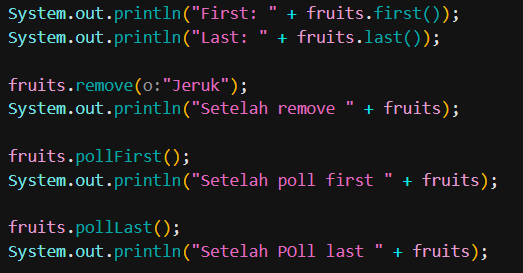


1. Compile dan run program. Mengapa urutan yang ditampilkan berbeda dengan urutan penambahan data ke dalam TreeSet fruits?

**Jawab:**

Karena TreeSet secara otomatis mengurutkan elemen berdasarkan urutan natural (natural ordering). Untuk tipe String, urutan ini adalah berdasarkan abjad (lexicographical order) sesuai Unicode

1. Tambahkan kode program sebagai berikut:

****

1. Apa yang dilakukan oleh method first(), last(), remove(), pollFirst(), dan pollLast()?

Jawab:

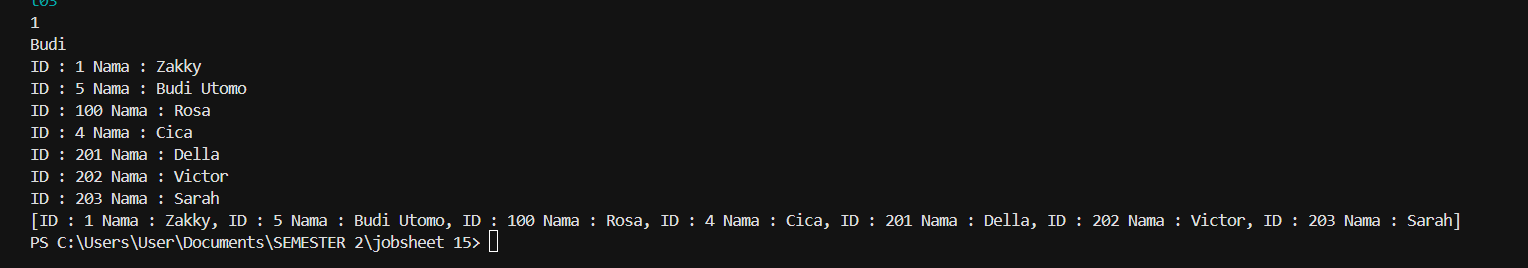
* 1. **First()** Mengambil elemen pertama (terkecil) dari TreeSet sesuai urutan yang terurut (ascending).
  2. **Last()** Mengambil elemen terakhir (terbesar) dari TreeSet.
  3. **Remove()** Mengambil elemen terakhir (terbesar) dari TreeSet.
  4. **pollFirst()** Menghapus dan mengembalikan elemen pertama dari TreeSet.
  5. **pollLast()** Menghapus dan mengembalikan elemen terakhir dari TreeSet.

**Kode program**

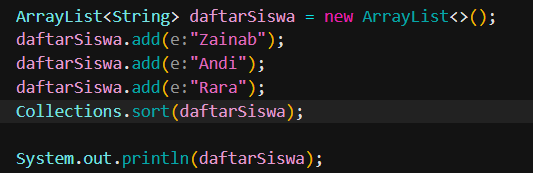
**Kode class TreeSetDemo03.java**

|  |
| --- |
| import java.util.TreeSet;  public class TreeSetDemo03 {      public static void main(String[] *args*) {          TreeSet<String> fruits = new TreeSet<>();          fruits.add("Mangga");          fruits.add("Apel");          fruits.add("jeruk");          fruits.add("Jambu");          for (String temp : fruits) {              System.out.println(temp);          }            System.out.println("First: " + fruits.first());          System.out.println("Last: " + fruits.last());          fruits.remove("Jeruk");          System.out.println("Setelah remove " + fruits);          fruits.pollFirst();          System.out.println("Setelah poll first " + fruits);          fruits.pollLast();          System.out.println("Setelah POll last " + fruits);      }  } |

**Hasil kode**

****

**Praktikum – Sorting**



**Kode program**

**Class Customer03.java**

|  |
| --- |
| public class Customer03 {      public int id;      public String name;      public Customer03(){      }      public Customer03(int *id*, String *name*) {          this.id = *id*;          this.name = *name*;      }      public String toString(){          return "ID : " + this.id + " Nama : " + this.name;      }  } |

**Class Book03.java**

|  |
| --- |
| public class Book03 {      public String isbn;      public String title;      public Book03(){      }      public Book03(String *isbn*, String *title*) {          this.isbn = *isbn*;          this.title = *title*;      }      public String toString(){          return "ISBN : " + this.isbn + " title : " + this.title;      }  } |

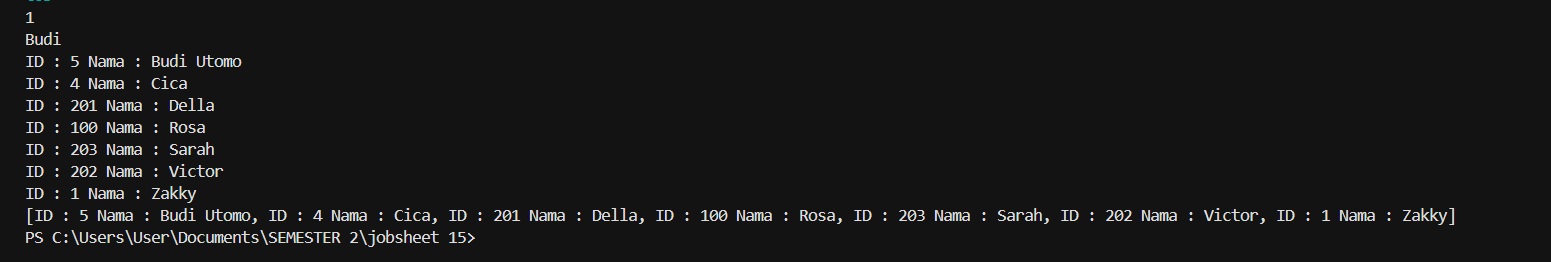
**Class TreeSetDemo03.java**

|  |
| --- |
| import java.util.ArrayList;  import java.util.Collections;  import java.util.TreeSet;  public class TreeSetDemo03 {      public static void main(String[] *args*) {          TreeSet<String> fruits = new TreeSet<>();          fruits.add("Mangga");          fruits.add("Apel");          fruits.add("jeruk");          fruits.add("Jambu");          for (String temp : fruits) {              System.out.println(temp);          }            System.out.println("First: " + fruits.first());          System.out.println("Last: " + fruits.last());          fruits.remove("Jeruk");          System.out.println("Setelah remove " + fruits);          fruits.pollFirst();          System.out.println("Setelah poll first " + fruits);          fruits.pollLast();          System.out.println("Setelah POll last " + fruits);          ArrayList<String> daftarSiswa = new ArrayList<>();          daftarSiswa.add("Zainab");          daftarSiswa.add("Andi");          daftarSiswa.add("Rara");          Collections.sort(daftarSiswa);          System.out.println(daftarSiswa);      }  } |

**Class DemoArrayList03.java**

|  |
| --- |
| import java.util.ArrayList;  public class DemoArrayList03 {      public static void main(String[] *args*) {            ArrayList<Customer03> customers = new ArrayList<>();          Customer03 customer1 = new Customer03(1, "Zakky");          Customer03 customer2 = new Customer03(5, "Budi");          customers.add(customer1);          customers.add(customer2);          customers.add(new Customer03(4, "Cica"));          customers.add(2, new Customer03(100, "Rosa"));          System.out.println(customers.indexOf(customer2));          Customer03 customer = customers.get(1);          System.out.println(customer.name);          customer.name= "Budi Utomo";          ArrayList<Customer03> newCustomers = new ArrayList<>();          newCustomers.add(new Customer03(201, "Della"));          newCustomers.add(new Customer03(202, "Victor"));          newCustomers.add(new Customer03(203, "Sarah"));          customers.addAll(newCustomers);          customers.sort((*c1*, *c2*) -> *c1*.name.compareTo(*c2*.name));            for (Customer03 cust : customers) {              System.out.println(cust.toString());          }            System.out.println(customers);      }  } |

**Hasil kode**

****